

# Rechtspolitik aus der Maschine

---

Dr. Dr. Hanjo Hamann

2021-08-16T15:41:47

Neue Impulse für die Politik und neue Gesetze können aus vielen Bereichen kommen; nicht zuletzt die Ökonomie wird gerne als „[most politically influential social science](#)“ bezeichnet. Die *Rechtspolitik* nimmt freilich einen spezifisch juristischen Blickwinkel ein, der durchaus interdisziplinär informiert sein kann, doch stets zumindest *auch* eine juristische Perspektive zu beinhalten hat, insbesondere eine solche, die aus der Kritik des geltenden Rechts erwächst.<sup>1</sup>

Inwiefern dann „Rechtspolitik aus der Maschine“? In unserem in der *Zeitschrift für Digitalisierung und Recht* erschienenen Beitrag,<sup>2</sup> der unlängst [auf diesem Blog besprochen](#) wurde, versuchen wir, den juristischen Blickwinkel durch informatische Methoden zu bereichern („aus der Maschine“ ist also nur als Hilfsmittel zu verstehen – wie ein guter Espresso, der trotz Herstellung „aus der Maschine“ viel Handwerkskunst bedarf).

Wie der Artikel im Titel bereits verrät, geschieht dies in unserer Anwendung mittels (schwacher) Künstlicher Intelligenz in Form eines probabilistischen Topic Models, das *grosso modo* anhand der Ähnlichkeiten verschiedener Texte diese in Sammlungen sortiert und innerhalb der Sammlungen die übereinstimmendsten Wörter ausgibt.<sup>3</sup> Die „Texte“ sind in diesem Fall alle deutschen Bundesgesetze (aufgearbeitet aus der [Kompilation von Séan Fobbe](#), für Details siehe den ZfDR-Beitrag und den entsprechenden [Online-Appendix](#)). Letztlich geht der Algorithmus damit einer juristischen Kernaufgabe nach: der sinnigen (d.h. oftmals logisch-systematischen) Einordnung einzelner Gesetze (und deren Inhalte) in einen größeren Kontext.

Diese Sammlung und Organisierung kann an sich auch für die Auslegung von Gesetzen genutzt werden, aufgrund des groben Blickwinkels jedoch allenfalls für manche Analogieschlüsse und größere Zusammenhänge wie etwa rechtsgebietsübergreifende Ähnlichkeiten.<sup>4</sup> Einstweilen ist der rechtspolitische Erkenntnisgewinn vielversprechender. Das Einordnen verschiedener Gesetze und deren Inhalte unter einen Schirm ist schließlich recht nah am Kodifikationsprozess angesiedelt, wie [in diesem Blog bereits zusammengefasst](#). Ob und inwieweit die computergenerierte Zuordnung Sinn ergibt, ist Metier juristischer Einschätzung. Doch kann die algorithmische Vorarbeit neue Impulse geben und möglicherweise anerkannte Gepflogenheiten wenigstens zwecks einer wertenden Überprüfung in Frage stellen.

Diese wertende, und insoweit notwendigerweise menschlich zu erfolgende, kernjuristische Aufgabe soll durch folgende Visualisierungen jede:r selbst anstellen dürfen, ohne dass wir insoweit die Interpretation (über die von uns indiziell erfolgte Benennung der jeweiligen Netzwerke hinaus) vorprägen wollen; unsere

Schlussfolgerungen für manche Topics sind schließlich bereits im ZfDR-Beitrag nachlesbar.

Hierbei sind zwei Modelle verfügbar. Beide sind als Netzwerk visualisiert – eine Technik, die in Verbindung mit der deutschen Rechtswissenschaft insbesondere mit der Pionierarbeit von [Dr. Corinna Coupette](#) in Verbindung steht (siehe dazu den Überblick in [Juristische Netzwerkforschung](#), jüngst aus anderer Feder auch im [AöR](#), um nur zwei Beispiele ohne Anspruch auf Vollständigkeit zu nennen). Netzwerke sind relationale Graphen, die Verbindungen („Kanten“) zwischen Datenpunkten („Knoten“) darstellen, und einer visuellen wie technischen (etwa mittels Methoden aus der Informatik und mathematischen Graphentheorie) Interpretation zugänglich sind. In unserem Beispiel bedeutet dies, dass sich die Knoten aus den deutschen Bundesgesetzen zusammensetzen (dargestellt mit Namen als „Punkt“) und die Kanten die Verweise, die von einem auf ein anderes Gesetz weisen (wie etwa § 8b Abs. 2 Nr. 11 HGB, der auf § 9 InsO verweist, was dazu führt, dass eine Kante zwischen HGB und InsO besteht; für Details und Grenzen dieser Verweisauslesung, die nur explizite externe Verweise umfasst, siehe den [Online-Appendix](#)).

Insgesamt konnten wir zwischen den 1.663 Bundesgesetzestexten 21.321 solche Verweise identifizieren, die im Netzwerk dargestellt werden (was auch erklärt, warum wir das Gesamtnetzwerk nicht darstellen, da es ohne Veränderung der Darstellungsform zu einem müßigen Suchbild führen würde). Das Gesamtnetzwerk kann durch verschiedene Algorithmen in Themensammlungen unterteilt werden. Die Ergebnisse zwei dieser Algorithmen sollen hier illustriert werden.

Das erste Modell entspricht dem des ZfDR-Beitrags. Es handelt sich um ein probabilistisches Topic Model, genauer gesagt der [Latent Dirichlet Allocation](#), erstellt mit der von [#eh##ek](#) bereitgestellten [gensim-Bibliothek](#) für Python. Es visualisiert die im Beitrag aufgelisteten zehn relevantesten Topics (Themensammlungen). Auf untenstehendem Bild, das eines der zehn bereitgestellten Topics darstellt, ist eine interaktive Version mit allen Topics verlinkt, bei der ferner einige Parameter eingestellt werden können (wie etwa Kantenlänge oder Knotenradius).

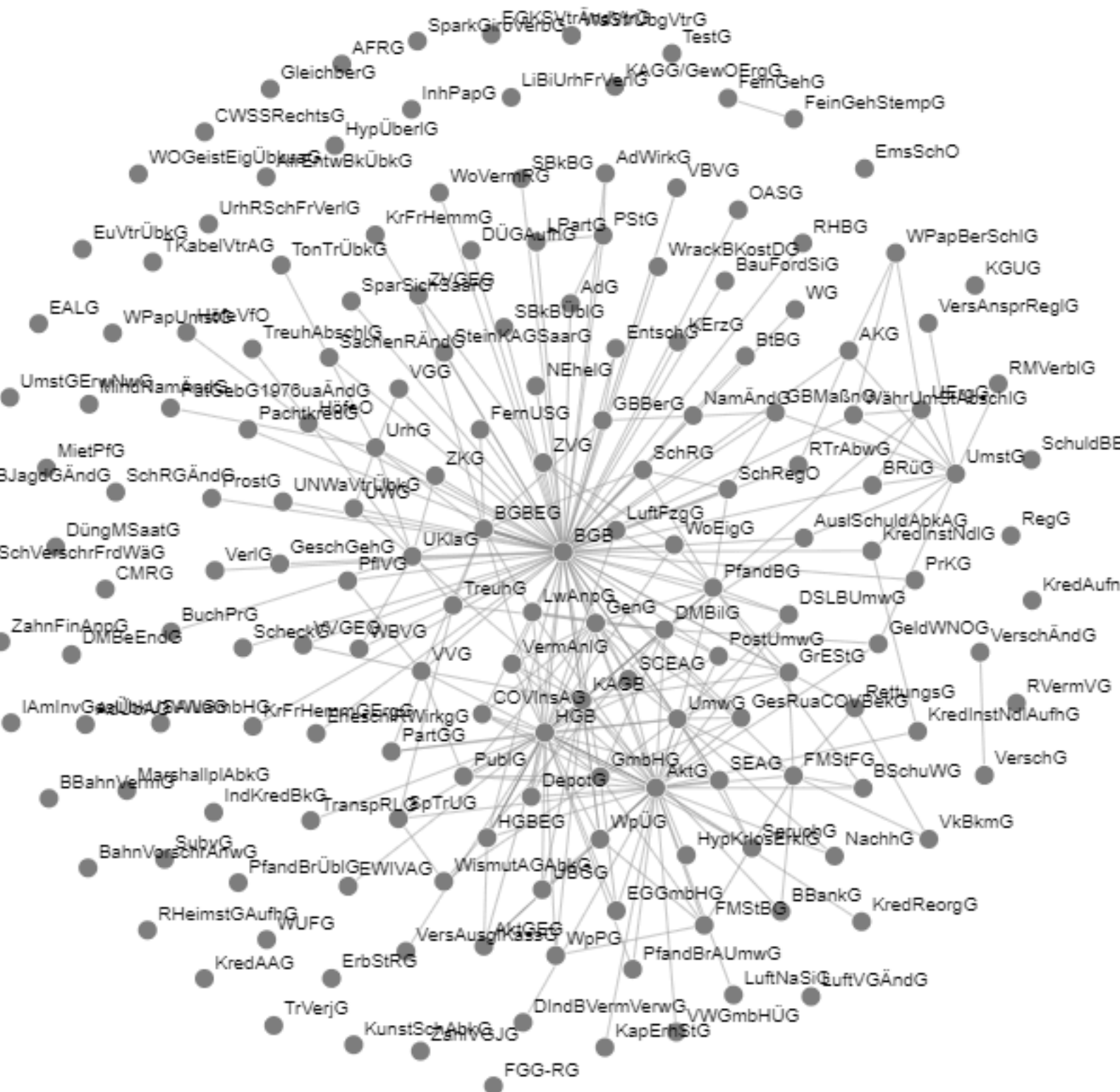


Abb. 1: [LDA-Modellierung](#) (verwendet in ZfDR 2021, S. 155 ff.)

Das zweite Modell ist ein spezifisch für diesen Blog-Beitrag ergänzter Alternativalgorithmus, dessen Funktionsweise eine grundlegend andere ist, der jedoch auch eine Form des unüberwachten Maschinellen Lernens anwendet (implementiert mit der Bibliothek [cdlib](#) von Rossetti). Während das Topic Model

nämlich von den Wörtern ausgeht und so die Texte auf Ähnlichkeiten untersucht, unterteilt der sog. [Leiden-Algorithm](#) als Weiterentwicklung des [Louvain-Modularity-Algorithm](#) nur anhand der Netzwerkstruktur. Dabei versucht er die sogenannte *Modularity*, die ein Maß für die Stärke des inneren Zusammenhangs einer Unterteilung in Cluster darstellt, zu maximieren. Dies geschieht ausgehend von einer zunächst uninteressanten Unterteilung, in der jedes Gesetz sein eigenes Cluster darstellt. Nun wird sukzessive untersucht, ob das Zusammenfügen von beliebigen Cluster-Paaren zu einem Anstieg in der *Modularity* führt. Dies geschieht so lange, bis kein Anstieg mehr durch Zusammenfügen erreicht werden kann. Anders als *Louvain* ist der *Leiden-Algorithm* insofern verfeinert, als er die als nächstes zusammenzufügenden Cluster systematisch anhand der bisherigen Aggregationen auswählt und so zu zusammenhängenden Ergebnissen als *Louvain* gelangt, der in bestimmten Fällen nicht zusammenhängende Unterteilungen generiert.<sup>5</sup> Wie beim ersten Bild ist unten eines der Cluster abgebildet und beim Klicken hierauf eine interaktiv konfigurierbare Version mit allen zehn bereitgestellten Clustern verlinkt.

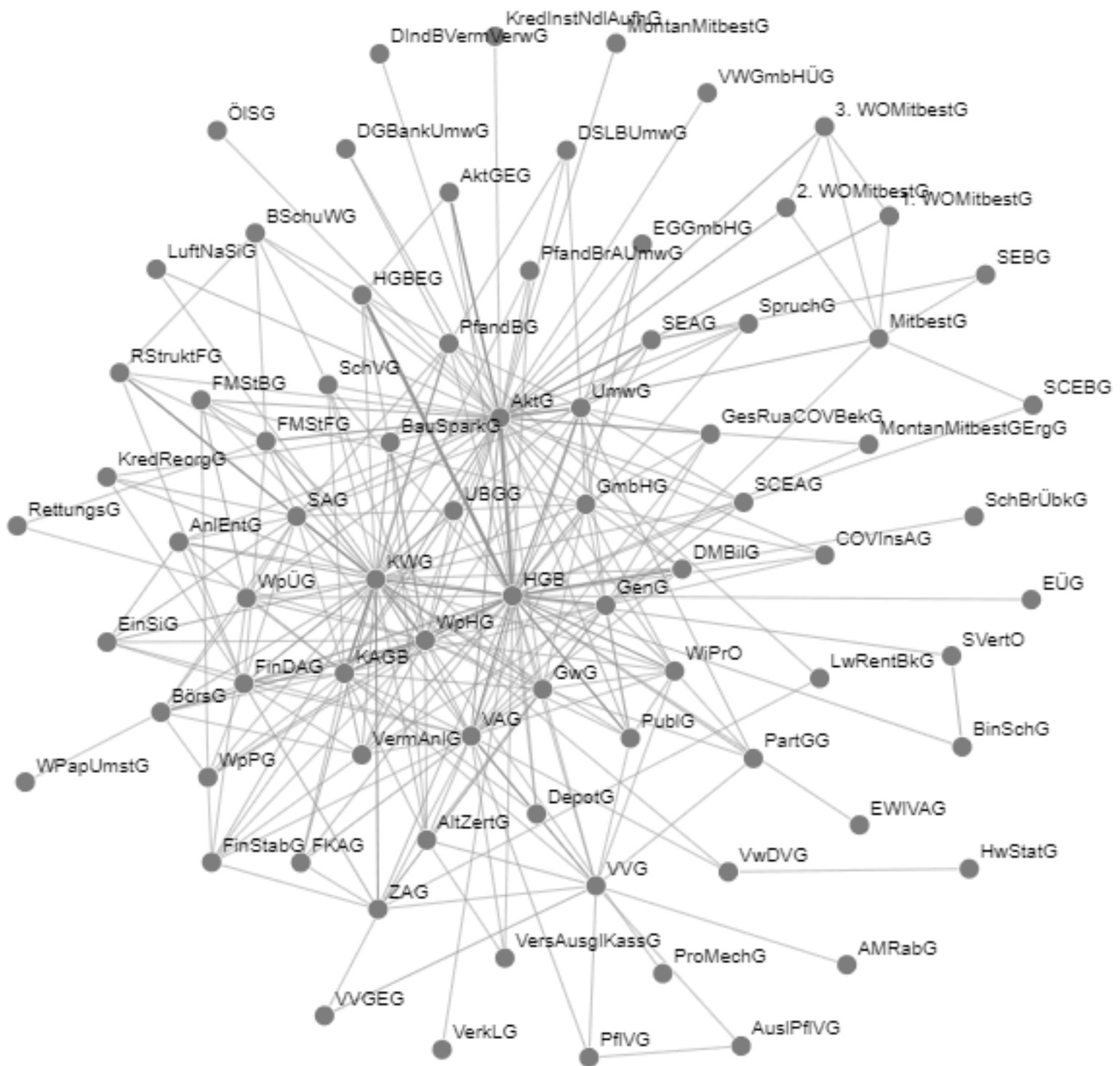


Abb. 2: Modellierung nach Leiden-Algorithm

Beide Visualisierungen erfolgten mit dem von *Wapman* open-source bereitgestellten Python-Paket [WebWeb](#). Unser Code für beide Visualisierungen ist als Ergänzung des [Online-Appendix](#) zum ZfDR-Beitrag abrufbar.

Eine genauere Deutung beider Visualisierungen möchten wir den geneigten Lesenden überlassen, und hoffen auf spannende Erkenntnisse für die „Rechtspolitik aus der Maschine“.

